BAB II

TINJAUAN PUSTAKA

2.1 *Data Mining*

Data mining adalah merupakan proses semi otomatis yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* untuk mengekstrasi dan mengidentifikasi informasi pengetahuan potensial dan berguna yang bermanfaat yang tesimpan di dalam database yang besar (Turban, E dkk : 2005). *Data mining* juga merupakan proses analisa yang dirancang untuk menelusuri data untuk mendapatkan bentuk konsisten dan hubungan yang sistematik antara variabel yang kemudian divalidasi dengan menggunakan sub set data yang baru (Xu, Shuxiang dan Ling Chen : 2008).

Secara umum, *data mining* adalah proses menambang pengetahuan dari sekumpulan data yang sangat besar, *data mining* merupakan suatu langkah dalam *knowledge discovery in database* (Han, Jiawei dkk : 2001).

Data mining juga didefinisikan sebagai serangkaian proses untuk menggali nilai tambah dari suatu kumpulan data berupa pengetahuan yang selama ini tidak diketahui secara manual dan merupakan analisis otomatis dari data yang berjumlah besar atau komples dengan tujuan untuk menemukan pola atau kecenderungan yang penting yang biasanya tidak disadari keberadaanya (Nilakant, K : 2011).

Pengumpulan data yang dilakukan di *data mining* adalah untuk mendapatkan data mentah yang dibutuhkan untuk transformasi data. Transformasi

data dilakukan untuk mengubah data mentah hasil dari pengumpulan data menjadi format yang dapat di proses oleh *data mining*.

2.2 *Decision Tree*

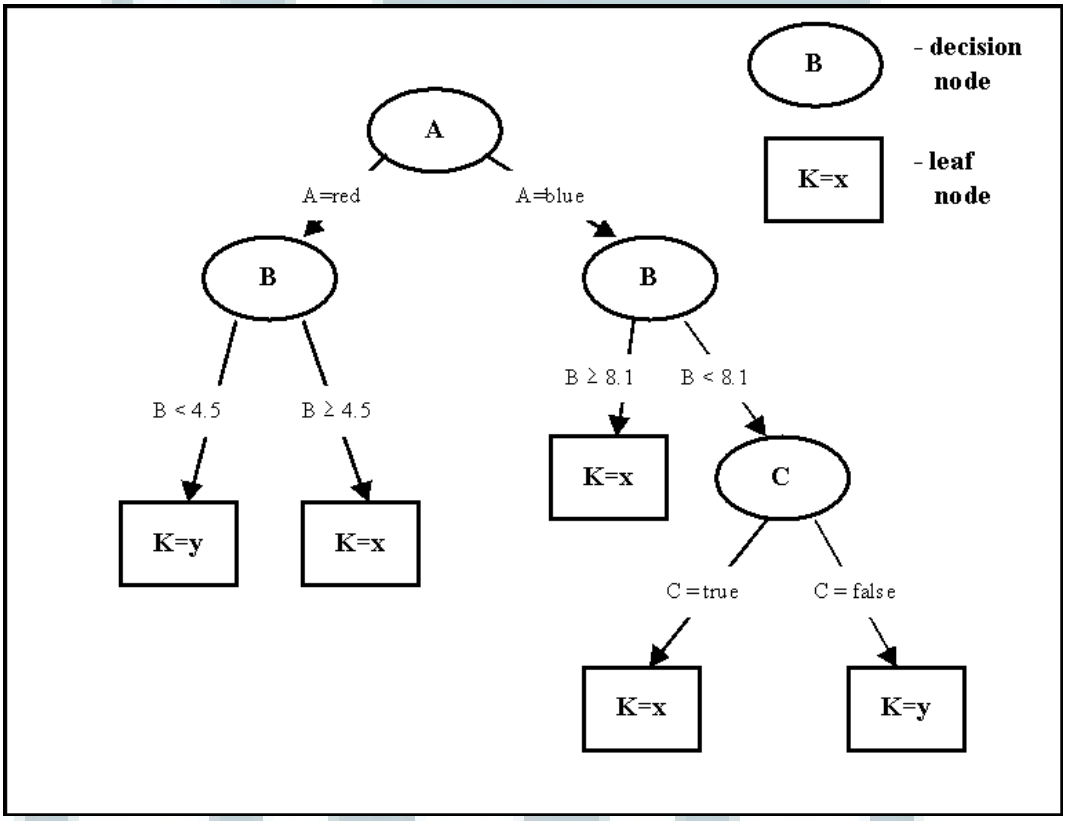
Menurut Larose, *decision tree* adalah salah satu metode yang digunakan untuk klasifikasi yang melibatkan konstruksi dari pohon yang terdiri dari *node* keputusan dan berhubungan dengan cabang – cabang dari *node* akar hingga *node* daun atau *node* terakhir. Pada *node* keputusan attribut akan diuji, dan setiap hasil akan menghasilkan cabang. Setiap cabang akan diarahkan ke *node* lain hingga *node* akhir untuk menghasilkan suatu keputusan (Larose, T. D : 2005).

Decision tree adalah struktur *flowchart* yang mempunyai *tree*, dimana setiap simpul internal menandakan suatu tes attribut, setiap cabang merepresentasikan hasil tes, dan simpul daun merepresentasikan kelas atau distribusi kelas (Sujana : 2010).

Manfaat dari penggunaan *decision tree* adalah memiliki kemampuan untuk mem-*break down* proses pengambilan keputusan yang kompleks menjadi lebih simpel sehingga pengambilan keputusan akan lebih menginterpretasikan solusi dari permasalahan. Pohon keputusan juga berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel input dengan sebuah variable target.

Pada proses mengklasifikasikan *sample* yang tidak diketahui, nilai atribut akan diuji pada *decision tree* dengan cara melacak jalur dari titik akar sampai titik akhir, kemudian akan diprediksi kelas yang ditempati *sample* baru tersebut. *Decision tree* mempunyai 3 tipe simpul yaitu (M, Saffi : 2011) :

1. Simpul akar dimana tidak memiliki cabang yang masuk dan memiliki cabang lebih dari satu, terkadang tidak memiliki cabang sama sekali.
2. Simpul *internal* dimana hanya memiliki 1 cabang yang masuk, dan memiliki lebih dari 1 cabang yang keluar.
3. Simpul daun atau simpul akhir dimana hanyan memiliki 1 cabang yang masuk, dan tidak memiliki cabang sama sekali.



Gambar 2.1 Contoh *Decision Tree*

Sumber (http://dms.irb.hr/tutorial/images/dtree_image.gif)

2.3 Algoritma C 4.5

Algoritma C 4.5 adalah salah satu algoritma yang digunakan untuk membuat suatu *decision* tree berdasarkan *data training* yang telah disiapkan.

Secara umum algoritma C 4.5 digunakan untuk membangun *decision tree* dengan cara sebagai berikut (Kusrini : 2007) :

1. Pilih atribut sebagai akar.
2. Buat cabang untuk masing – masing nilai.
3. Bagi kasus dalam cabang.
4. Ulangi proses untuk masing – masing cabang sampai semua kasus selesai.

Pemilihan atribut yang akan digunakan sebagai akar, didasarkan pada nilai Gain tertinggi dari atribut – atribut yang ada. Hal pertama yang perlu dilakukan untuk menentukan akar adalah menghitung Entrophy dari data yang ada .Entrophy bisa dikatakan sebagai kebutuhan *bit* untuk menyatakan suatu kelas. Semakin tinggi Entrophynya semakin baik digunakan dalam mengekstrasi suatu kelas (Adyama, Aang : 2013) . Entrophy dapat dihitung dengan rumus (Korting, Thales Sehn) :

$$Entropy(S) = \sum_{i=1}^{n} - pi * log_2\ pi$$

.....Rumus 2.1 Rumus Entrophy

Sumber (Korting, Thales Sehn)

Keterangan :

S = Himpunan Kasus

A = Fitur

n = Jumlah partisi S

pi = Proporsi dari Si terhadap S

Setelah menghitung Entrophy barulah kita dapat menghitung Gain untuk penentuan akar dan cabang – cabang dari *decision tree* berdasarkan pada Gain terbesar. Gain dapat di hitung dengan rumus (Korting, Thales Sehn) :

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{Si}{S} * Entropy(Si)$$

.....Rumus 2.2 Rumus Gain

Sumber (Korting, Thales Sehn)

Keterangan :

S = Himpunan Kasus

A = Atribut

N = Jumlah Partisi Atribut A

Si = Jumlah Kasus pada Partisi Ke – i

S = Jumlah Kasus dalam S

Berikut ini adalah *psuedo code* untuk algoritma C 4.5 (Upadhayay, Anurag dkk):

In pseudo code the algorithm is:
1. Check for base cases
2. For each attribute *a* (. Find the normalized information gain from splitting on *a)*
3. Let *a_best* be the attribute with the highest normalized information gain

4. Create a decision *node* that splits on *a_best*
5. Recur on the sublists obtained by splitting on *a_best*, and add those nodes as children of *node*

Gambar 2.2 *Psuedo code* Algoritma C 4.5

Sumber (Upadhayay, Anurag dkk)

Berikut ini adalah contoh kasus keputusan bermain basket yang diambil dari bahan mata kuliah *data mining* di STMIK AMIKOM Yogyakarta.Luthfi(2013) :

Tabel 2.1 Tabel keputusan bermain basket

Sumber (Luthfi(2013))

No	OUTLOOK	TEMPERATURE	HUMIDITY	WINDY	PLAY
1	Sunny	Hot	High	FALSE	No
2	Sunny	Hot	High	TRUE	No
3	Cloudy	Hot	High	FALSE	Yes
4	Rainy	Mild	High	FALSE	Yes
5	Rainy	Cool	Normal	FALSE	Yes
6	Rainy	Cool	Normal	TRUE	Yes
7	Cloudy	Cool	Normal	TRUE	Yes
8	Sunny	Mild	High	FALSE	No
9	Sunny	Cool	Normal	FALSE	Yes
10	Rainy	Mild	Normal	FALSE	Yes
11	Sunny	Mild	Normal	TRUE	Yes
12	Cloudy	Mild	High	TRUE	Yes
13	Cloudy	Hot	Normal	FALSE	Yes
14	Rainy	Mild	High	TRUE	No

Dari tabel 2.1 di atas dapat ditunjukan contoh sampel dari *attribute* apa saja yang menentukan hasil keputusan bermain basket, sehingga keputusan bermain didapatkan.

Setelah mendapatkan sampel seperti yang ditunjukan oleh tabel 2.1 maka penghitungan entrophy dan gain dapat dilakukan. Dengan rumus yang telah dijelaskan sebelumnya. Berikut contoh perhitungan entrophy dari sampel di atas.

$$\text{Entropy(Total)} = (-\tfrac{4}{14} * \log_2(\tfrac{4}{14})) + (-\tfrac{10}{14} * \log_2(\tfrac{10}{14}))$$

$$\text{Entropy(Total)} = 0.863120569$$

Gambar 2.3 Contoh penghitungan Entrophy

Sumber (Luthfi(2013))

Berikut adalah contoh perhitungan gain *outlook* :

$$\text{Gain(Total, Outlook)} = \text{Entropy(Total)} - \sum_{i-1}^{n} \frac{|Outlook|}{|Total|} * \text{Entropy(Outlook)}$$

$$\text{Gain(Total, Outlook)} = 0.863120569 - ((\tfrac{4}{14} * 0) + (\tfrac{5}{14} * 0.723) + (\tfrac{5}{14} * 0.97))$$

$$\text{Gain(Total, Outlook)} = 0.23$$

Gambar 2.4 Contoh perhitungan gain

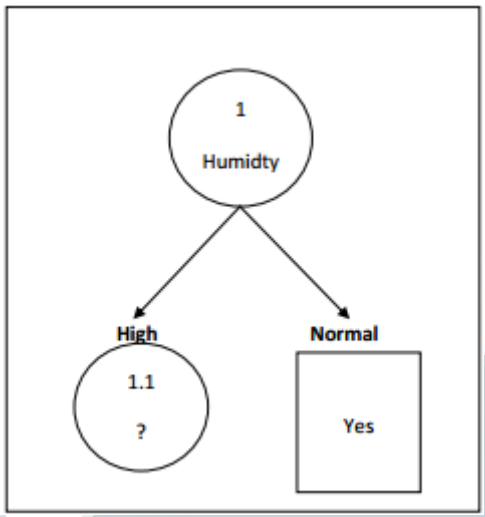
Sumber (Luthfi(2013))

Tabel 2.2 Perhitungan *Node*

Sumber (Luthfi(2013))

Node			Jumlah kasus (S)	Tidak (S1)	Ya (S2)	Entropy	Gain
1	TOTAL		14	4	10	0.863120569	
	OUTLOOK						0.258521037
		CLOUDY	4	0	4		
		RAINY	5	1	4	0.721928095	
		SUNNY	5	3	2	0.970950594	
	TEMPERATURE						0.183850925
		COOL	4	0	4	0	
		HOT	4	2	2	1	
		MILD	6	2	4	0.918295834	
	HUMIDITY						0.370506501
		HIGH	7	4	3	0.985228136	
		NORMAL	7	0	7	0	
	WINDY						0.000597
		FALSE	4	2	2	1	
		TRUE	3	2	1	0.918295834	

Dari tabel 2.2 ditunjukan hasil perhitungan semua gain berdasarkan entrophy total sehingga dapat di tentukan mana node yang akan menjadi akar dari pohon. Penentuan akar pada pohon ditentukan berdasarkan *node* dengan nilai gain tertinggi pada contoh sampel ini ditunjukan oleh *humidity* dengan nilai gain 0.370506501. *Node* yang tidak memiliki hasil diantara salah satu S1 atau S2 maka nilai entrophynya dipastikan 0 atau dipastikan tidak ada cabang, karena tidak menghasilkan salah satu dari S1 atau S2 yang artinya sudah memiliki hasil keputusan. Pada kasus ini nilai *humidity* pada saat normal maka dipastikan akan menghasilkan *Ya*.. Dan berikut adalah hasil pohon dengan *humidity* sebagai akarnya.



Gambar 2.5 Akar hasil perhitungan gain

Sumber (Luthfi(2013))

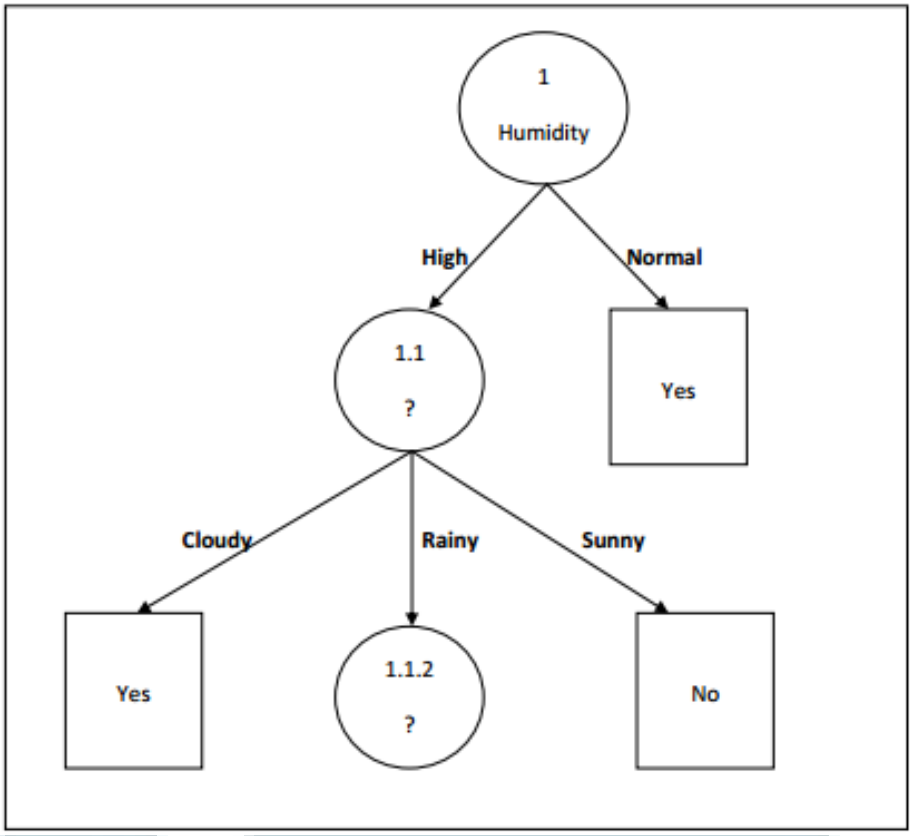
Untuk mendapatkan *node* berikutnya (simpul *internal*) penghitungan gain akan didasarkan oleh entrophy *node* akarnya atau *node* sebelumnya. Pada *node* berikutnya ditentukan bahwa *outlook* adalah *node* dengan nilai gain tertinggi. Berikut adalah table pehitungan node terakhir :

Tabel 2.3 Penghitungan node 1.1.2

Sumber (Luthfi(2013))

Node			Jumlah kasus (S)	Tidak (S1)	Ya (S2)	Entropy	Gain
1.1.2	HUMIDITY-HIGH dan OUTLOOK-RAINY		2	1	1		
	TEMPERATURE						0
		COOL	0	0	0	0	
		HOT	0	0	0	0	
		MILD	2	1	1	1	
	WINDY						1
		FALSE	1	0	1	0	
		TRUE	1	1	0	0	

Setelah ditentukan bahwa *outlook* merupakan cabang berikutnya, maka akan dilihat apakah *attribute outlook* yakni *cloudy*, *rainy*, *sunny* udah memiliki keputusan, jika sudah maka pohon keputusan akhir akan terbentuk. Maka dari hasil sampel dan penghitungan gain dari semua *node*, maka dapat dihasilkan sebuah pohon keputusan sebagai berikut :



Gambar 2.6 Hasil akhir dari pohon keputusan

Sumber (Luthfi(2013))